



AYUDANTÍA 12: Repaso Arreglos Bidimensionales o Matrices de Dos Dimensiones

IIC1102 – Introducción a la Programación – Sección 4

PROBLEMAS

1. 13 - Primer Semestre 2001 - Pregunta 2 (modificado)

Dado una matriz cuadrada de $N \times N$, como por ejemplo alguna de las siguientes:

$$\begin{array}{l} A = \begin{array}{cccc} 4 & 4 & 5 & 4 \\ 4 & 4 & 4 & 4 \\ 3 & 4 & 4 & 4 \\ 8 & 4 & 1 & 4 \end{array} \\ B = \begin{array}{ccc} 6 & 6 & 6 \\ 6 & 6 & 6 \\ 7 & 7 & 7 \end{array} \\ C = \begin{array}{cc} 1 & 2 \\ 2 & 1 \end{array} \end{array}$$

que se puede almacenar en un arreglo `int[][] matriz`.

a) Escriba el método `public static int filycol(int numero, int[][] matriz)` que tome como parámetros un número y una matriz, y verifique cuántas filas y cuántas columnas tienen en su respectiva fila o columna solamente el número especificado.

Por ejemplo:

Para la matriz **A** con el número **4**, la respuesta sería **3** pues tiene el número **4** en **una fila** y **dos columnas**.

Para la matriz **B** y el número **6**, la respuesta sería **2** pues tiene el número **6** en **dos filas**; para el número **7**, la respuesta sería **1** pues tiene el número **7** en **una fila**; y la respuesta es **0** para todos los otros números.

Para la matriz **C** la respuesta es **0** para cualquier número, pues no hay ninguna fila, ni columna donde se repita un número.

b) Escriba ahora el método `public static int diagonal(int numero, int[][] matriz)` que tome como parámetros un número entero y una matriz, y verifique cuántas diagonales (de las dos posibles) tienen ese número.

Por ejemplo, la matriz **A** con el número **4**, tiene una diagonal con dicho número y la respuesta sería **1**. Para la matriz **B**, la respuesta es **0** para cualquier número. Para la matriz **C** tanto para el número **1** como para el número **2** la respuesta es **1**, pues cada diagonal tiene sólo ese número.

2. Basado en 13 - Primer Semestre 2004 - Pregunta 2

Una imagen se puede representar como una matriz de dos dimensiones, en donde cada posición almacena un color.

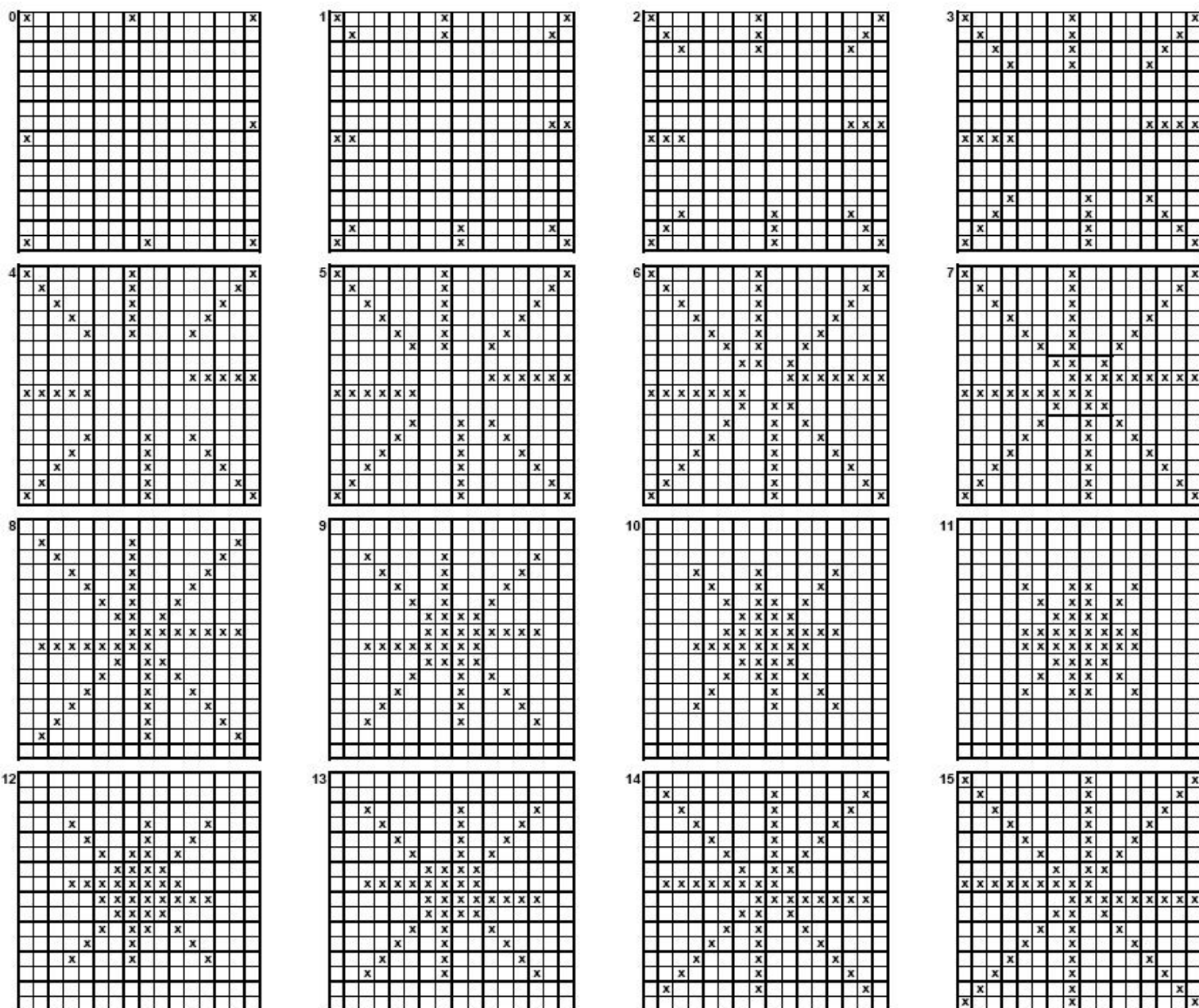
Un video corresponde a una serie de imágenes que se proyectan consecutivamente de forma muy rápida. Una forma muy simplificada de representar cada cuadro de una película almacenada digitalmente es a través de una matriz de dos dimensiones. De esta manera, un video digital se puede interpretar como un arreglo de matrices de dos dimensiones (o una matriz tridimensional*), donde el arreglo (o la primera dimensión) representa la aparición del cuadro dentro de la película (tiempo) y las matrices de dos dimensiones (o las dos dimensiones restantes) representan un cuadro de la película.

Este concepto permite representar, por ejemplo, una película en blanco y negro utilizando un arreglo de matrices (o una matriz tridimensional) de números enteros que almacena el número **0** en cada casilla de un cuadro donde el pixel es **negro** o un número **255** si en dicho casillero hay un pixel **blanco**.

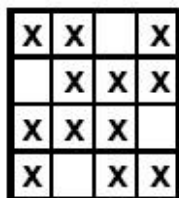
* En este curso **no** se verán matrices tridimensionales.

Hoy en día existen avanzados algoritmos para la búsqueda de patrones en videos digitales. En su versión más simple, el objetivo de dichos algoritmos es encontrar una cierta imagen en algún cuadro de los que componen la película.

Por ejemplo, suponga que se tiene el siguiente video de 16 cuadros, en donde la **x** corresponde a un **0** y los espacios vacíos corresponden a un **255**:



Y se desea buscar la siguiente imagen dentro del video:



En dicho caso, el algoritmo debería retornar el número 7 (ver la primera figura), ya que es el primer cuadro donde aparece la imagen buscada.

Para resolver este ejercicio, Ud. cuenta con la siguiente clase:

```
public class Imagen {
    private int[][] img;
    public Imagen(int[][] img) {
        this.img = img;
    }
    public int[][] getArreglo() {
        return img;
    }
}
```

A continuación deberá crear la clase **Pelicula**, que tendrá un atributo **private Imagen[] peli** donde se almacenarán los cuadros de la película. Además:

a) Escriba el siguiente método de la clase **Pelicula**:

```
public boolean sonIguales(int numCuadro, int i, int j, Imagen img);
```

Donde **numCuadro** es el número de un cuadro dentro de la película.
Las variables **i** y **j** son las coordenadas dentro del cuadro número **numCuadro** donde se quiere saber si se encuentra la imagen **img**.

Para que el método tenga sentido, el tamaño de la imagen **img** debe ser menor o igual al tamaño de los cuadros de la película.

Deberá retornar **true** si la imagen es igual al segmento de la película cuyo vértice superior izquierdo está definido por las coordenadas anteriores, o **false** en caso contrario.

b) Escriba el siguiente método de la clase **Pelicula**:

```
public int buscarImagen(Imagen img);
```

Deberá retornar el número del cuadro de la película donde primero aparece la imagen **img** o **-1** si no la encontró.